

setting the synchronization access mode of the write pointer to be either normal or sync to effect the behavior of adding data to the buffer.

39. The method of claim 38 wherein the fetching and adding includes executing a fetch and add operation.

40. The method of claim 38 wherein when the synchronization access mode of the write pointer is set to normal, the storing includes overwriting data previously stored in the buffer and not yet read.

41. The method of claim 38 wherein when the synchronization access mode of the write pointer is set to sync, the storing includes waiting to overwrite data previously stored in the buffer until the data has been read.

42. The method of claim 38 wherein the setting of the synchronization access mode of the write pointer is transparent to the adding of the data to the buffer.

43-56. (Cancelled)

57. (New) A method in a computer system for accessing data, the method comprising:

defining a pointer to point to a location within the data, the pointer having a synchronization access mode;

setting the synchronization access mode of the pointer to be either unsynchronized or synchronized to effect a desired behavior of accessing the data;

allowing accessing code to access the data using the pointer with the current setting of the synchronization access mode;

changing the synchronization access mode of the pointer to effect the desired behavior of accessing the data; and

allowing accessing code to access the data using the pointer with the changed setting of the synchronization access mode.

58. (New) The method of claim 57 wherein the setting of the synchronization access mode is changed to change the desired behavior of accessing the data.

59. (New) The method of claim 57 wherein when the synchronization access mode is set to unsynchronized, access to the location within the data is permitted regardless of whether location is empty or full.

60. (New) The method of claim 57 wherein when the synchronization access mode is set to sync, read access to the location within the data is permitted only when the location is full.

61. (New) The method of claim 60 after when the read access, the location is set to empty.

62. (New) The method of claim 57 wherein when the synchronization access mode is set to sync, write access to the location within the data is permitted only when the location is empty.

63. (New) The method of claim 62 wherein after the write access, the location is set to full.

64. (New) The method of claim 57 wherein the behavior of the accessing code is modified without modifying the accessing code.

65. (New) The method of claim 57 including storing in forwarding locations at located past an end of the data pointers to locations at the other end of the data, the pointers having forwarding enabled so that when a forwarding location is accessed, the access is directed to the pointed to location at the other end of the data.

66. (New) The method of claim 57 including storing a pointer to an invalid location in a location adjacent to the data with forwarding of that location enabled so that when the location adjacent to the data is accessed an exception is raised.

67. (New) The method of claim 57 wherein the data is accessed by multiple readers and writers.

68. (New) The method of claim 57 wherein the data is accessed by multiple producers.

69. (New) The method of claim 57 wherein the data is accessed by multiple consumers.

70. (New) The method of claim 57 including
when access to the location within the data by a thread is blocked,
enabling an exception to be raised when the location is next accessed;
and
blocking the thread; and
when an exception is raised as a result of access by another thread to that location,
completing the access by that other thread to that location; and
restarting execution of the blocked thread.

71. (New) The method of claim 70 wherein when access by the thread to the location is blocked, saving a state of the thread and storing a reference to the thread in the location.

72. (New) The method of claim 71 wherein the reference is a pointer to a data structure that identifies the blocked thread and the saved state.

73. (New) The method of claim 72 wherein the data structure indicates the value that was stored in the location before storing the reference.

74. (New) A method in a computer system for reading from a buffer of data, the method comprising:

defining a pointer to point to a location within the buffer, the pointer having a synchronization access mode;

setting the synchronization access mode of the pointer to be either normal or sync to effect a desired behavior of accessing the buffer; and

under control of accessing code, reading from a location within the buffer using the pointer wherein the behavior of the accessing code depends on the setting of the synchronization access mode of the pointer and can be changed without modifying the accessing code.

75. (New) The method of claim 74 wherein the setting of the synchronization access mode is changed to change the desired behavior of accessing the buffer.

76. (New) The method of claim 74 wherein when the synchronization access mode is set to normal, the reading from the location within the buffer is permitted when the location is empty.

77. (New) The method of claim 74 wherein when the synchronization access mode is set to sync, the reading from the location within the buffer is permitted only when the location is full.

78. (New) The method of claim 74 wherein when the synchronization access mode is set to sync, writing to the location within the buffer is permitted only when the location is empty.

79. (New) The method of claim 74 including storing in forwarding locations at located past an end of the buffer pointers to locations at the other end of the buffer, the pointers having forwarding enabled so that when a forwarding location is accessed, the access is directed to the pointed to location at the other end of the buffer.

80. (New) The method of claim 74 including storing a pointer to an invalid location in a location adjacent to the buffer with forwarding of that location enabled so that when the location adjacent to the buffer is accessed an exception is raised.

81. (New) The method of claim 74 including
 when reading from the location within the buffer by a thread is blocked,
 enabling an exception to be raised when the location is next accessed via
 a write access; and
 blocking the thread; and
 when an exception is raised as a result of a write access to that location by
 another thread,
 completing the write access to that location by that other thread; and
 restarting execution of the blocked thread to complete the reading from
 the location.

82. (New) A method in a computer system for writing to a buffer of data, the method comprising:
 defining a pointer to point to a location within the buffer, the pointer having a
 synchronization access mode;
 setting the synchronization access mode of the pointer to be either normal or
 sync to effect a desired behavior of accessing the buffer; and
 under control of accessing code, writing to a location within the buffer using the
 pointer wherein behavior of the accessing code depends on the setting of
 the synchronization access mode of the pointer and can be changed
 without modifying the accessing code.

83. (New) The method of claim 82 wherein the setting of the synchronization access mode is changed to change the desired behavior of accessing the buffer.

84. (New) The method of claim 82 wherein when the synchronization access mode is set to normal, the writing to the location within the buffer is permitted when the location is full.

85. (New) The method of claim 82 wherein when the synchronization access mode is set to sync, the writing to the location within the buffer is permitted only when the location is empty.

86. (New) The method of claim 82 wherein when the synchronization access mode is set to sync, reading from the location within the buffer is permitted only when the location is full.

87. (New) The method of claim 82 including storing in forwarding locations located past an end of the buffer pointers to locations at the other end of the buffer, the pointers having forwarding enabled so that when a forwarding location is accessed, the access is directed to the pointed to location at the other end of the buffer.

88. (New) The method of claim 82 including storing a pointer to an invalid location in a location adjacent to the buffer with forwarding of that location enabled so that when the location adjacent to the buffer is accessed an exception is raised.

89. (New) The method of claim 82 including
 when writing by a thread to the location is blocked,
 enabling an exception to be raised when the location is next accessed via
 a read access; and
 blocking the thread; and
 when an exception is raised as a result of a read access by another thread to
 that location,
 completing the read accessing from that location by that other thread; and
 restarting execution of the blocked thread to complete the writing to the
 location.

90. (New) A computer system for accessing data, comprising:
 a buffer of data;
 a read pointer and a write pointer that point to locations within the buffer, each
 pointer having a synchronization access mode being set to either

unsynchronized or synchronized to effect a desired behavior of accessing the buffer of data;

a plurality of accessing programs that access the buffer of data using the read and write pointers; and

means for changing the setting of the synchronization access mode of a pointer to effect a change in the desired behavior of accessing the buffer of data without modifying the accessing programs.

91. (New) The computer system of claim 90 the desired behavior of accessing the data is changed by changing the setting of the synchronization access mode of the pointers.

92. (New) The computer system of claim 90 wherein when the synchronization access mode of the pointers is set to unsynchronized, access to the location within the data is permitted regardless of whether location is empty or full.

93. (New) The computer system of claim 90 wherein when the synchronization access mode of the read pointer is set to synchronized, read access to the location within the data is permitted only when the location is full.

94. (New) The computer system of claim 90 wherein when the synchronization access mode of the write pointer is set to synchronized, write access to the location within the data is permitted only when the location is empty.

95. (New) The computer system of claim 90 wherein each accessing program operates in a different thread.

96. (New) The computer system of claim 90 including storing in forwarding locations located past an end of the buffer pointers to locations at the other end of the data, the pointers having forwarding enabled so that when a forwarding location is accessed, the access is directed to the pointed to location at the other end of the buffer.

97. (New) The computer system of claim 90 including storing a pointer to an invalid location in a location adjacent to the buffer with forwarding of that location enabled so that when the location adjacent to the buffer is accessed an exception is raised.

98. (New) The computer system of claim 90 wherein the data is accessed by multiple reading threads and writing threads.

99. (New) The computer system of claim 90 including
when access by a thread to a location pointed to by a pointer is blocked,
enabling an exception to be raised when the location is next accessed;
and
blocking the thread; and
when an exception is raised as a result of access by another thread to that location,
completing the access by that other thread to that location; and
restarting execution of the blocked thread.

100. (New) A computer system for accessing data, comprising:
means for storing data within a buffer;
means for pointing to locations within the buffer, the pointing means having a synchronization access mode;
means for changing the synchronization access mode of the pointing means;
and
means for accessing the data of the buffer using the pointing means wherein the accessing behavior of the pointing means.

101. (New) The computer system of claim 100 wherein the accessing means includes multiple producers and multiple consumers of the data within the buffer.

102. (New) The computer system of claim 100 wherein the pointing means includes a read pointer and a write pointer.